

Familia Profesional Informática y Telecomunicaciones		Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma	
Centro Educativo IES Campanillas (sede PTA)		Módulo Profesional Programación Código: 0485 N.º de créditos ECTS: 14	Profesor Luis José Sánchez González
Curso lectivo: 2021 / 2022	Grupo: 1º DAM	Trimestre: Segundo – Control 1	Fecha: 15 de febrero de 2022

INSTRUCCIONES

- ➔ El alumno debe entregar una carpeta con las soluciones al examen cuyo nombre debe estar formado por "Ex" seguido del número de lista, seguido de las iniciales. Por ejemplo, Facundo Romuedo Piladro que es el número 8 de la lista entregaría una carpeta con nombre **Ex08frp**.
- ➔ Los ficheros o carpetas correspondientes a las soluciones se deben nombrar igual que la carpeta junto con el número del ejercicio, por ejemplo **Ex08frp1.java, Ex08frp2.java, etc.**
- ➔ En los comentarios de cada programa **se debe indicar el nombre completo**, la fecha y - si procede - el modelo.
- ➔ Únicamente se necesita entregar el código fuente en java, **no se deben entregar los archivos con la extensión .class**.

ARRAYS

1. Realiza un programa que genere un array cuadrado con números aleatorios entre el 1 y el 99, ambos incluidos. La longitud del lado del array (un entero mayor o igual a 1) se debe pedir al usuario. A continuación, el usuario podrá extraer un trozo de ese array indicando la fila y la columna donde se empieza a extraer y la longitud del lado del trozo.
Tanto el array original como el trozo se deben mostrar con los números perfectamente alineados, como se muestra en los ejemplos.
Las filas y las columnas se numeran empezando a contar en la 0.

Ejemplo 1:

Introduzca la longitud del lado del array: 8

```
37 68 42  8 57 11 77 94
89 94 61 99 77 25 48 79
10 87 15 37 76 77 79 47
88 37 39 47  4 87 15 22
96 69 29 16 60 76 38 19
40 85 61  8 16 35 36 48
39 60  2 54 55 97 18 54
72 37 56 12 76 51 89 64
```

Vamos a extraer un trozo de ese array.

Introduzca la fila para comenzar a extraer: 0

Introduzca la columna para comenzar a extraer: 0

Introduzca la longitud del lado que se va a extraer: 4

```
37 68 42  8
89 94 61 99
10 87 15 37
88 37 39 47
```

Ejemplo 2:

Introduzca la longitud del lado del array: 6

```
62 49  8 80 46 13
32 46 38 87 91 52
13 43 75 93 23 44
33 19 42 54 26 43
90 59 84 73 14 97
15  5 25 82 59 11
```

Vamos a extraer un trozo de ese array.

Introduzca la fila para comenzar a extraer: 2

Introduzca la columna para comenzar a extraer: 4

Introduzca la longitud del lado que se va a extraer: 2

```
23 44
26 43
```

Ejemplo 3:

Introduzca la longitud del lado del array: 5

```
36 91 80 99 41
84 87 25  5 91
28 31  3 61 29
89  3 86  2 86
 2 26 95 81 84
```

Vamos a extraer un trozo de ese array.

Introduzca la fila para comenzar a extraer: 0

Introduzca la columna para comenzar a extraer: 2

Introduzca la longitud del lado que se va a extraer: 3

```
80 99 41
25  5 91
 3 61 29
```

FUNCIONES

2. a) Crea la función **línea** para pintar líneas de caracteres que tienen un determinado carácter en los extremos y otro carácter de relleno. Prueba esta función en un programa con, al menos, los ejemplos que se proporcionan a continuación.

```
public static String linea(int longitud, char extremo, char relleno)
```

Ejemplos:

```
línea(5, '*', ' ') devuelve "*   *"
línea(5, '*', '*') devuelve "*****"
línea(6, 'X', '-') devuelve "X---X"
línea(3, '&', '$') devuelve "&$&"
línea(2, '&', '$') devuelve "&&"
línea(1, '&', '$') devuelve "&"
línea(0, '&', '$') devuelve ""
línea(-3, '&', '$') devuelve ""
```

- b) Utiliza esta función para pintar una pirámide hueca cuya altura se pide por teclado. Podemos suponer que el usuario va a introducir un número mayor o igual a 1.

Ejemplo 1:

Introduzca la altura de la pirámide: 8

```
      *
     * *
    *  *
   *   *
  *    *
 *     *
*      *
*      *
*****
```

Ejemplo 2:

Introduzca la altura de la pirámide: 2

```
      *
     * *
    * * *
```

Ejemplo 3:

Introduzca la altura de la pirámide: 1

```
      *
```

PROGRAMACIÓN ORIENTADA A OBJETOS

3. Se quiere implementar una aplicación para la gestión de un acuario. El primer paso será modelar las clases necesarias. Crea las clases **AnimalDeAcuario** (no instanciable), **Pez** y **Estrella** estableciendo la jerarquía correctamente. De todos los animales del acuario se debe conocer su **nombre** y **color**. Además, se debe conocer la **raza** de los peces y el **número de brazos** de las estrellas. Si un animal acuático se aparea con otro, nace un animalillo de la misma especie con las características de los padres cogidas de forma aleatoria. Por ejemplo, si se aparean dos estrellas de mar, una se llama Alicia y la otra Alfonso, el nombre de la cría podría ser Alicia o Alfonso. Si Alicia tiene 5 brazos y Alfonso tiene 6 brazos, la cría puede tener 5 o 6 brazos pero no otra cantidad. Con los peces sucede lo mismo.

Contenido del main:

```
Pez pepe = new Pez("Pepe", "azul", "globo");
Pez casilda = new Pez("Casilda", "naranja", "tropical");
System.out.println(pepe);
System.out.println(casilda);

Estrella lola = new Estrella("Lola", "violeta", 6);
Estrella rodolfo = new Estrella("Rodolfo", "verde", 7);
System.out.println(lola);
System.out.println(rodolfo);

// Apareamiento
Pez pececillo = pepe.seApareaCon(casilda);
Estrella estrellita = lola.seApareaCon(rodolfo);
System.out.println(pececillo);
System.out.println(estrellita);
```

Salida:

```
Soy Pepe, de color azul, un pez globo
Soy Casilda, de color naranja, un pez tropical
Soy Lola, de color violeta, una estrella con 6 brazos
Soy Rodolfo, de color verde, una estrella con 7 brazos
Soy Pepe, de color azul, un pez tropical
Soy Rodolfo, de color violeta, una estrella con 6 brazos
```

4. Se van a utilizar las clases **AnimalDeAcuario** y **Pez** creadas en el ejercicio anterior por lo que deberás hacer una copia de dichas clases en la carpeta correspondiente a este ejercicio. Además de hacer la copia, crea la clase **Pecera**. De una pecera se debe saber su **nombre** o identificador y los **litros** de capacidad. En una pecera se pueden meter peces. Para determinar la cantidad máxima de peces que caben en una pecera, se ha de tener en cuenta que no se deben meter más de 4 peces por litro de capacidad. En cualquier momento se podrá mostrar información sobre la pecera: nombre, litros de capacidad, número de peces que contiene y un listado con todos ellos.

Contenido del main:

```
Pez pepe = new Pez("Pepe", "azul", "globo");
Pez casilda = new Pez("Casilda", "naranja", "tropical");

Pecera miPecerilla = new Pecera("Pecerilla", 2);
miPecerilla.mete(pepe);
miPecerilla.mete(casilda);

for (int i = 0; i < 6; i++) {
    miPecerilla.mete(new Pez("Sara", "amarillo", "guppy"));
}

System.out.println(miPecerilla);
miPecerilla.mete(new Pez("Intruso", "rosa", "gurami"));
```

Salida:

```
Pecerilla de 2 litros contiene 8 peces:
Soy Pepe, de color azul, un pez globo
Soy Casilda, de color naranja, un pez tropical
Soy Sara, de color amarillo, un pez guppy
Soy Sara, de color amarillo, un pez guppy
Soy Sara, de color amarillo, un pez guppy
Soy Sara, de color amarillo, un pez guppy
Soy Sara, de color amarillo, un pez guppy
Soy Sara, de color amarillo, un pez guppy

Pecera llena, no se pueden meter más peces.
```

Nota: Con el fin de acortar el tiempo necesario para la realización del examen, no es necesario comprobar que los datos introducidos por el usuario son correctos y están en el rango adecuado. No obstante, hay que tener presente que en los ejercicios de clase y en el mundo real es muy importante validar siempre los datos.